Assignment No. 2

1.1 Title:

Consider a suitable dataset. For clustering of data instances in different groups, apply different clustering techniques (minimum 2) Eg. DBSCAN, X-Means. Visualize the clusters using suitable tool.

1.2 Problem Definition:

Visulize the Cluster using Suitable tool

1.3 Prerequisite:

- Basic concepts of ETL.

1.4 Software Requirements:

- RapidMiner

1.5 Hardware Requirement:

- PIV, 2GB RAM, 500 GB HDD, Lenovo A13-4089Model.

1.6 Learning Objectives:

Use RapidMiner to create K-means Clustering models and hierarchical clustering models (DBSCAN, X-Means)

1.7 Outcomes:

Visualize the effectiveness of the K-means Clustering algorithm and hierarchical clustering using graphic capabilities in RapidMiner

1.8 Theory Concepts:

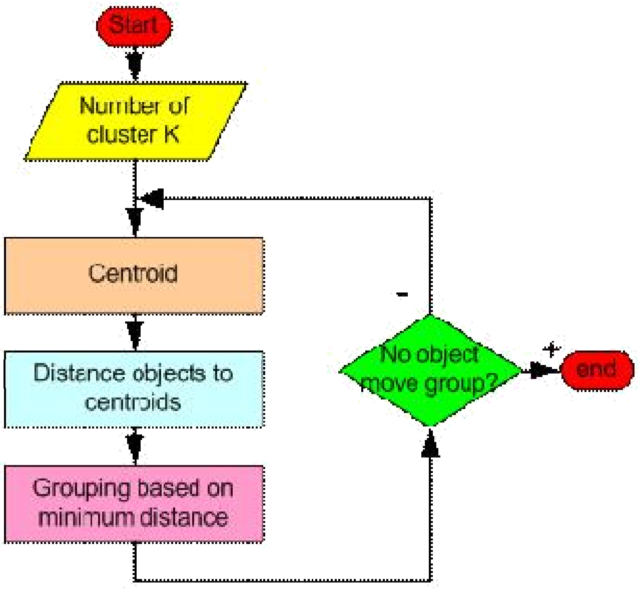
What is K-means clustering?

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable *K*. The algorithm works iteratively to assign each data point to one of *K* groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the *K*-means clustering algorithm are:

1. The centroids of the *K* clusters, which can be used to label new data
2. Labels for the training data (each data point is assigned to a single cluster)

Rather than defining groups before looking at the data, clustering allows you to find and analyze the groups that have formed organically. The "Choosing K" section below describes how the number of groups can be determined. Each centroid of a cluster is a collection of feature values which define the resulting groups. Examining the centroid feature weights can be used to qualitatively interpret what kind of group each cluster represents.

Steps to Perform K-Means Clustering


Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

As a simple illustration of a k-means algorithm, consider the following data set consisting of the scores of two variables on each of seven individuals:

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

This data set is to be grouped into two clusters. As a first step in finding a sensible initial partition, let the A & B values of the two individuals furthest apart (using the Euclidean distance measure), define the initial cluster means, giving:

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

The remaining individuals are now examined in sequence and allocated to the cluster to which they are closest, in terms of Euclidean distance to the cluster mean. The mean vector is recalculated each time a new member is added. This leads to the following series of steps:

	Cluster 1		Cluster 2	
Step	Individual	Mean Vector (centroid)	Individual	Mean Vector (centroid)
1	1	(1.0, 1.0)	4	(5.0, 7.0)
2	1, 2	(1.2, 1.5)	4	(5.0, 7.0)
3	1, 2, 3	(1.8, 2.3)	4	(5.0, 7.0)
4	1, 2, 3	(1.8, 2.3)	4, 5	(4.2, 6.0)
5	1, 2, 3	(1.8, 2.3)	4, 5, 6	(4.3, 5.7)
6	1, 2, 3	(1.8, 2.3)	4, 5, 6, 7	(4.1, 5.4)

Now the initial partition has changed, and the two clusters at this stage having the following characteristics:

	Individual	Mean Vector (centroid)
Cluster 1	1, 2, 3	(1.8, 2.3)
Cluster 2	4, 5, 6, 7	(4.1, 5.4)

But we cannot yet be sure that each individual has been assigned to the right cluster. So, we compare each individual's distance to its own cluster mean and to that of the opposite cluster. And we find:

Individual	Distance to Mean (centroid) of Cluster 1	Distance to Mean (centroid) of Cluster 2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8 .	0.6
7	2.8	1.1

Only individual 3 is nearer to the mean of the opposite cluster (Cluster 2) than its own (Cluster 1). In other words, each individual's distance to its own cluster mean should be smaller that the distance to the other cluster's mean (which is not the case with individual 3). Thus, individual 3 is relocated to Cluster 2 resulting in the new partition:

	Individual	Mean Vector (centroid)
Cluster 1	1, 2	(1.3, 1.5)
Cluster 2	3, 4, 5, 6, 7	(3.9, 5.1)

The iterative relocation would now continue from this new partition until no more relocations occur. However, in this example each individual is now nearer its own cluster mean than that of the other cluster and the iteration stops, choosing the latest partitioning as the final cluster solution.

Steps

1. Set working directory
2. Get data from datasets
3. Execute the model
4. View the output
5. Plot the results

DBSCAN Clustering:

DBSCAN (for density-based spatial clustering of applications with noise) is a density-based clustering algorithm because it finds a number of clusters starting from the estimated density distribution of corresponding nodes.

The cluster formed, which is a subset of the points of the data set, satisfies two properties: All points within the cluster are mutually density-connected. If a point is density-connected to any point of the cluster, it is part of the cluster as well.

DBSCAN requires two parameters:

1. Epsilon
2. minimum number of points required to form a cluster (minPts).

DBSCAN starts with an arbitrary starting point that has not been visited.

If a point is found to be a dense part of a cluster, its epsilon-neighborhood is also part of that cluster. Hence, all points that are found within the epsilon-neighborhood are added, as is their own epsilon-neighborhood when they are also dense. This process continues until the density-connected cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.

If no id attribute is present, this operator will create one. The 'Cluster 0' assigned by DBSCAN operator corresponds to points that are labeled as noise. These are the points that have less than *min points* points in their epsilon-neighborhood.

1.9 Outputs

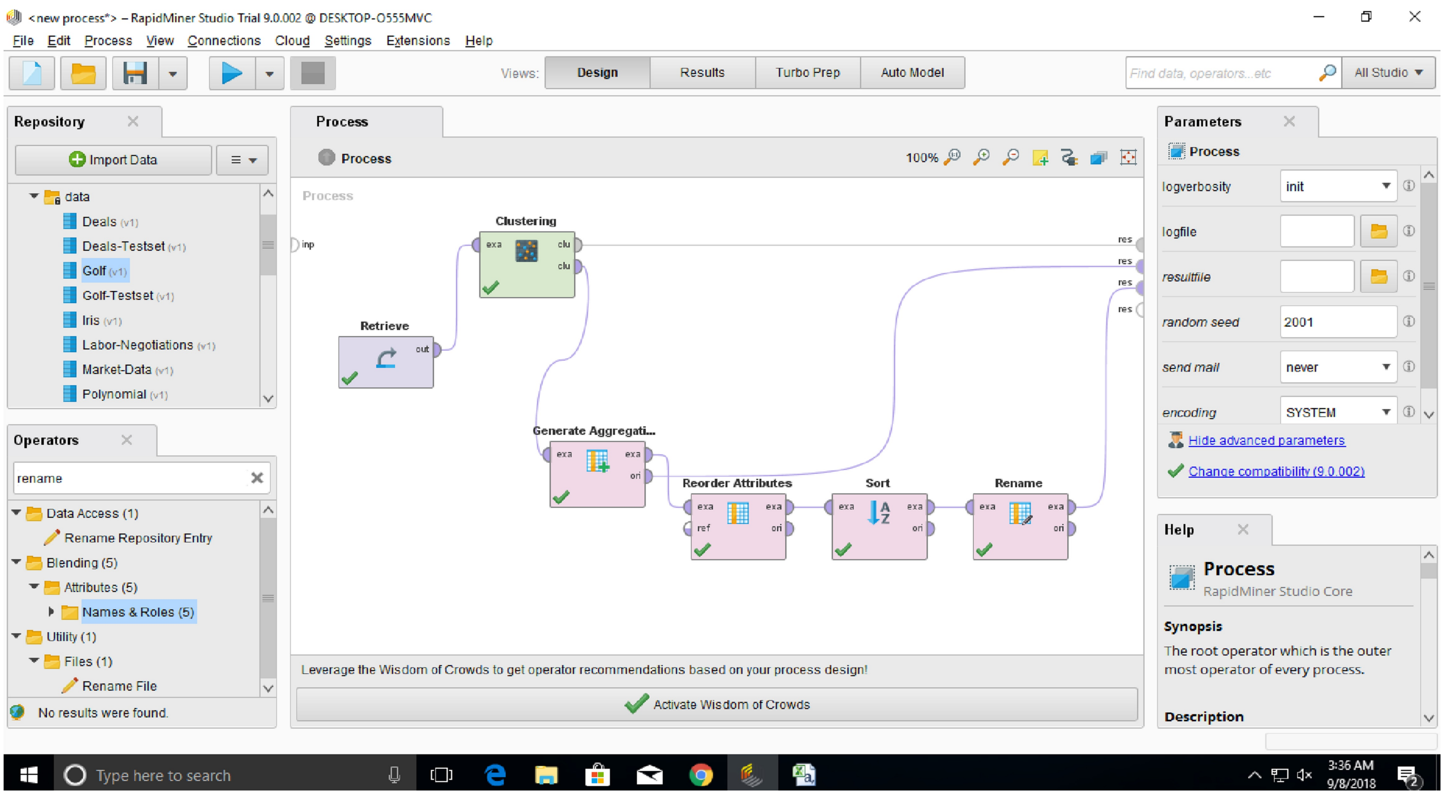
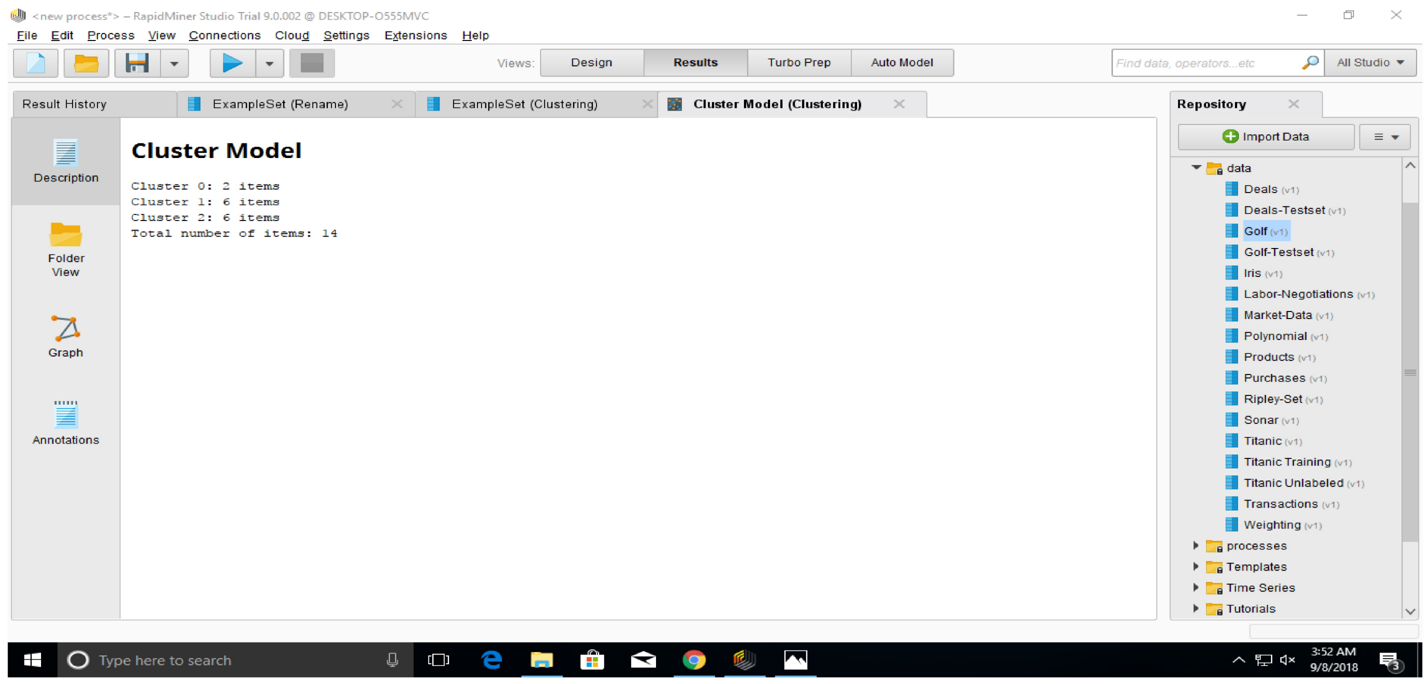


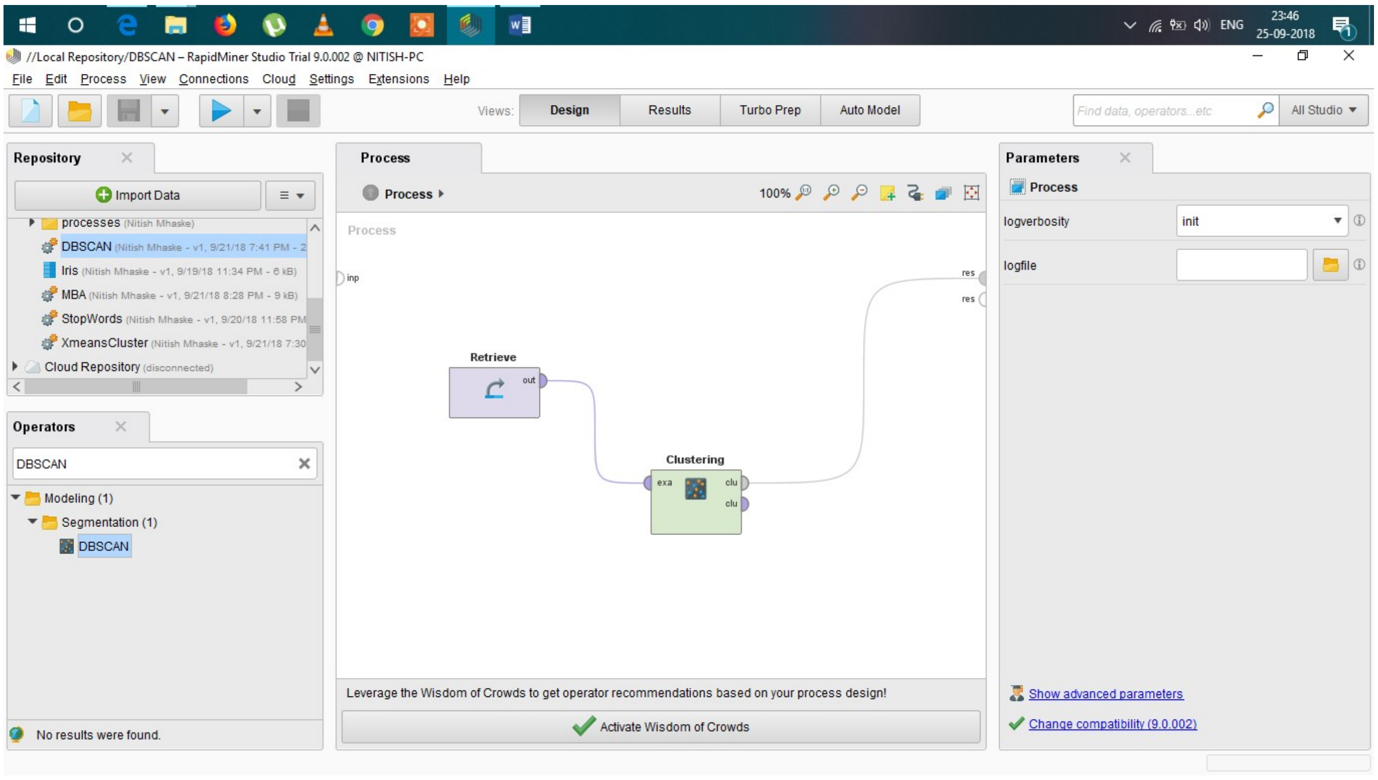
Figure 2.1 K-Means Clustering



Cluster Model

```
Cluster 0: 2 items
Cluster 1: 6 items
Cluster 2: 6 items
Total number of items: 14
```

Figure 2.2 K-Means Cluster


//Local Repository/DBSCAN – RapidMiner Studio Trial 9.0.002 @ NITISH-PC
File Edit Process View Connections Cloud Settings Extensions Help
Views: Design Results Turbo Prep Auto Model
Repository
Import Data
Operators
Modeling (1)
Segmentation (1)
DBSCAN
Process
Retrieve
Clustering
Parameters
Show advanced parameters
Change compatibility (9.0.002)
Activate Wisdom of Crowds

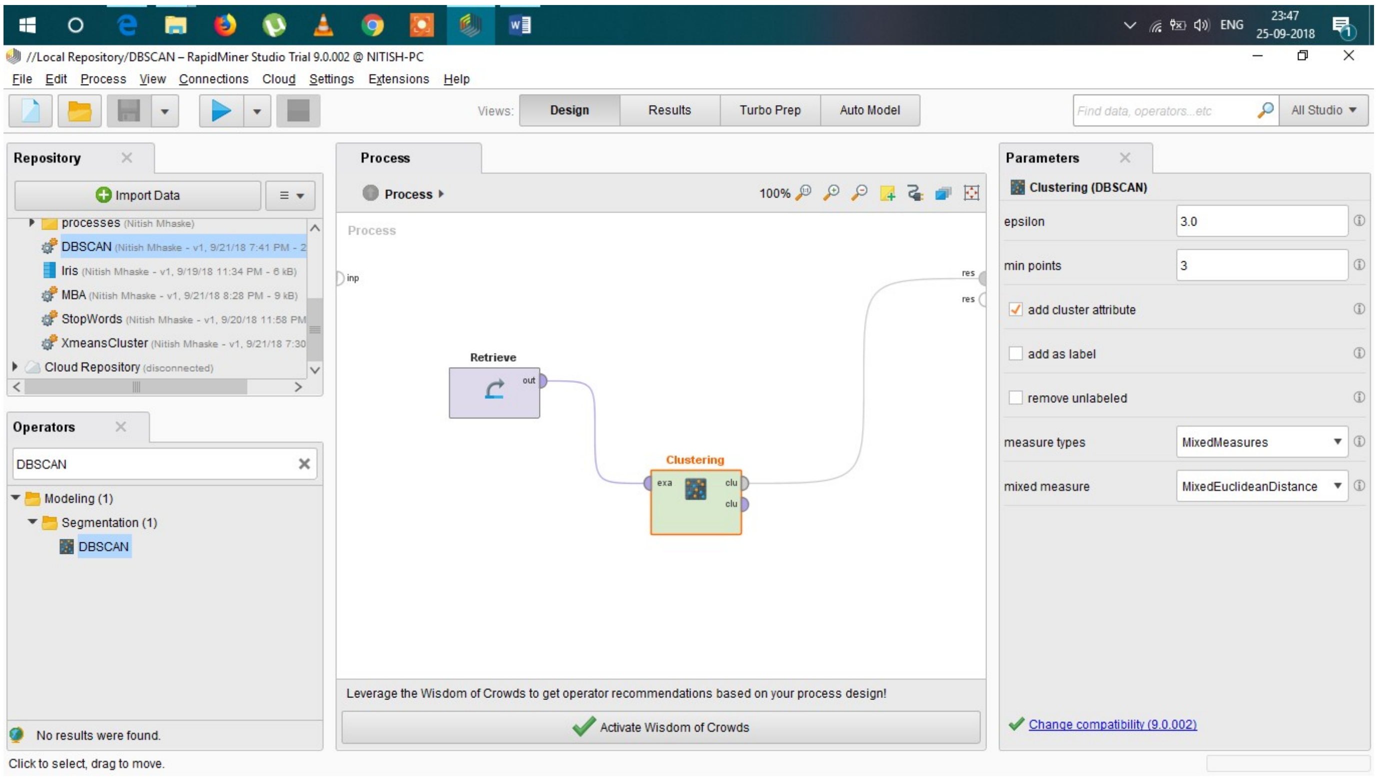
processes (Nitish Mhaske)
DBSCAN (Nitish Mhaske - v1, 9/21/18 7:41 PM - 2
Iris (Nitish Mhaske - v1, 9/19/18 11:34 PM - 6 kB)
MBA (Nitish Mhaske - v1, 9/21/18 8:28 PM - 9 kB)
StopWords (Nitish Mhaske - v1, 9/20/18 11:58 PM
XmeansCluster (Nitish Mhaske - v1, 9/21/18 7:30
Cloud Repository (disconnected)

No results were found.

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Process	
logverbosity	init
logfile	

Figure 2.2 DBSCAN Clustering


//Local Repository/DBSCAN – RapidMiner Studio Trial 9.0.002 @ NITISH-PC
File Edit Process View Connections Cloud Settings Extensions Help
Views: Design Results Turbo Prep Auto Model
Repository
Import Data
Operators
Modeling (1)
Segmentation (1)
DBSCAN
Process
Retrieve
Clustering
Parameters
Change compatibility (9.0.002)
Activate Wisdom of Crowds

No results were found.

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Click to select, drag to move.

Clustering (DBSCAN)	
epsilon	3.0
min points	3
add cluster attribute	✓
add as label	
remove unlabeled	
measure types	MixedMeasures
mixed measure	MixedEuclideanDistance

Figure 2.2 DBSCAN Clustering (*epsilon,min_point selection*)

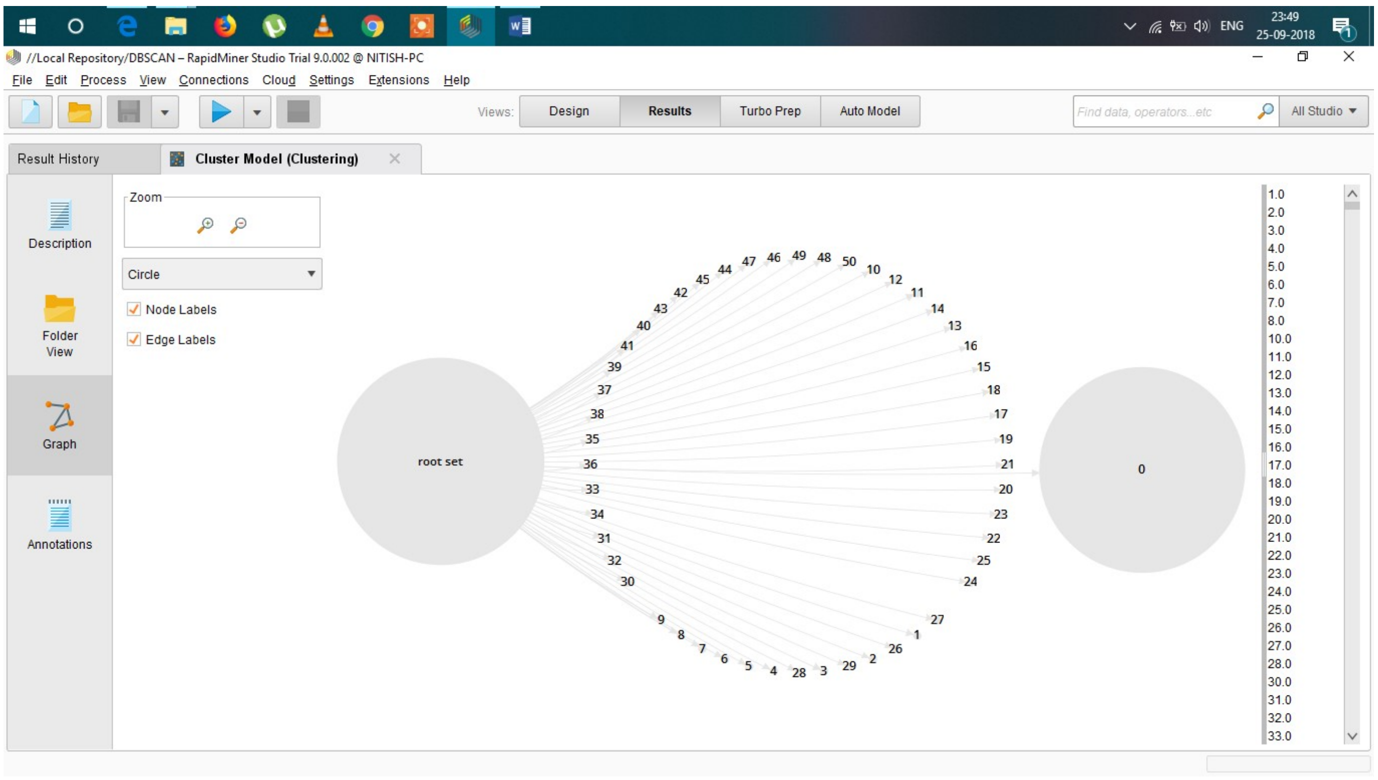

//Local Repository/DBSCAN – RapidMiner Studio Trial 9.0.002 @ NITISH-PC
File Edit Process View Connections Cloud Settings Extensions Help
Views: Design Results Turbo Prep Auto Model
Find data, operators...etc
All Studio
Result History
Cluster Model (Clustering)
Description
Folder View
Graph
Annotations
Zoom
Circle
Node Labels
Edge Labels
root set
0
23:49
25-09-2018
ENG

Figure 2.2 DBSCAN Clustering (*Circular Cluster Representation*)

1.10 Conclusion:

At the end of this assignment, we visualize the effectiveness of the K-means Clustering algorithm and DBSCAN clustering using graphic capabilities in RapidMiner

References

1 http://www.statmethods.net/advstats/cluster.html
2 http://people.revoledu.com/kardi/tutorial/Clustering/Numerical%20Example.htm
3 http://www.stat.berkeley.edu/~s133/Cluster2a.html
4 http://www.rdatamining.com/examples/kmeans-clustering
5 https://www.youtube.com/watch?v=zAS5vjrXCvY
6 https://docs.rapidminer.com/downloads/RapidMiner-v6-user-manual.pdf